



REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mohamed Khider – BISKRA

Faculté des Sciences Exactes, des Sciences de la Nature et de la Vie

Département d'informatique

N° d'ordre : 04/IVA/M2/2023

Mémoire

Présenté pour obtenir le diplôme de master académique en

Informatique

Parcours : **Image et Vie Artificielle (IVA)**

Colorisation des images médicales par deep learning

Par :

BOUREBATE HAMZA

Soutenu le 21/06/2023 devant le jury composé de :

Babahenini Mohamed Chaouki	Professeur	Président
Djerou Leila	Professeur	Rapporteur
Ababsa Tarek	M.C.B	Examineur

Année universitaire 2022-2023

Remerciements

Alhamdoulillah qui m'a facilité mon périple, et qui m'a fait sortir des ténèbres de l'ignorance à la lumière de la science et de la connaissance, merci mon dieu de m'avoir donné la capacité d'écrire et de réfléchir, la force d'y croire, la patience d'aller jusqu'au bout du rêve et le bonheur de lever mes mains vers le ciel, Tout le mérite revient à Dieu avant tout.

Le travail présenté dans cette mémoire a été réalisé au département d'informatique de la faculté FSES NV de l'Université de Mohamed kheider Biskra, sous la direction **Pr. DJEROU LEILA**, je voudrais saisir cette occasion et exprimer ma plus grande gratitude pour mon encadreur pour sa disponibilité et la confiance qu'il m'a accordée.

Je tiens également à remercier les membres du **jury** de m'avoir fait l'honneur de juger mon travail.

À mon très cher **père** « De tous les pères, tu es le meilleur.

Grâce à toi papa j'ai appris le sens du travail et de la responsabilité, Tes conseils ont toujours guidé mes pas vers la réussite, Je tiens à honorer l'homme que tu es, En ce jour, j'espère réaliser l'un de tes rêves... »

Et particulièrement à ma très chère **maman** qui a toujours été là pour moi. Ta prière et ta bénédiction m'ont été d'un grand secours pour mener à bien mes études et tout au long de ma vie.

À mon frère «**Mohamed Abd esammed**» À mes sœurs «**Samiha, Miada et Amina**».

A ma chère épouse **Ahlem**, qui a été très patiente avec moi et a pris beaucoup de son temps, surtout pendant sa grossesse. A mon fils **Mohamed Dia Eddin**.

Un merci spécial à mon directeur au travail, **Abd elhay chalouai**, pour toutes les installations et le soutien qui m'ont permis de concilier travail et études, ainsi que pour les encouragements et le soutien.

Enfin, je remercie mes collègues de travail, en particulier au service informatique «**Oussama, Lakhder et Sallah**», pour le soutien qu'ils m'ont apporté pendant mes études et la préparation de ce travail

Résumé

La colorisation des images médicales joue un rôle important dans l'amélioration de l'interprétation des images médicales en niveaux de gris et la précision du diagnostic, et dans cette mémoire nous nous intéressons à la colorisation des images tomographiques du corps humain.

Le travail essentiel de notre système est de former un réseau neuronal convolutif (CNN) qui prédit les couleurs des images en niveaux de gris en formant un modèle sur une base de données d'images de tomographie du corps humain.

En conclusion, cette mémoire contribue au domaine de la colorisation des images médicales en introduisant un nouveau paradigme d'apprentissage en profondeur qui améliore efficacement l'interprétabilité et la précision diagnostique des images médicales en niveaux de gris. La recherche démontre le potentiel des techniques d'apprentissage en profondeur pour révolutionner l'analyse et la colorisation des images médicales et fournit des informations précieuses pour le développement d'outils avancés en imagerie médicale et en diagnostic.

Mots-clés : image de tomodensitométrie (CT), deep learning, réseaux de neurones convolutifs (CNN).

ABSTRACT

The colorization of medical images plays an important role in improving the interpretation of grayscale medical images and the accuracy of diagnosis, and in this thesis we are interested in the colorization of tomographic images of the human body.

The essential job of our system is to train a convolutional neural network (CNN) that predicts the colors of grayscale images by training a model on a database of human body tomography images.

In conclusion, this thesis contributes to the field of medical image colorization by introducing a new deep learning paradigm that effectively improves the interpretability and diagnostic accuracy of grayscale medical images. The research demonstrates the potential of deep learning techniques to revolutionize the analysis and colorization of medical images and provides valuable insights for the development of advanced tools in medical imaging and diagnostics.

Keywords : computed tomography (CT) image, deep learning, convolutional neural networks (CNN).

ملخص:

يلعب تلوين الصور الطبية دورًا مهمًا في تحسين تفسير الصور الطبية ذات التدرج الرمادي ودقة التشخيص، وفي هذه الأطروحة نهتم بتلوين الصور المقطعية لجسم الإنسان.

تتمثل المهمة الأساسية لنظامنا في تدريب شبكة عصبية تلافيفية تتنبأ بألوان الصور الرمادية من خلال تدريب نموذج على قاعدة بيانات لصور التصوير المقطعي لجسم الإنسان.

في الختام، تساهم هذه الرسالة في مجال تلوين الصور الطبية من خلال تقديم نموذج جديد للتعلم العميق يعمل على تحسين قابلية تفسير الصور الطبية ذات التدرج الرمادي ودقتها التشخيصية بشكل فعال. يوضح البحث إمكانات تقنيات التعلم العميق لإحداث ثورة في تحليل وتلوين الصور الطبية ويوفر رؤى قيمة لتطوير أدوات متقدمة في التصوير الطبي والتشخيص.

الكلمات المفتاحية: صورة التصوير المقطعي، التعلم العميق، الشبكات العصبية التلافيفية.

Table des matières

Introduction Générale	12
I Colorisation des images	14
I.1 Introduction	15
I.2 Définition	15
I.3 Domaine d’application de colorisation	18
I.3.1 Colorisation des films :	18
I.3.2 Colorisation des images médicales	19
I.3.3 Compression d’image couleur	20
I.4 Les approches de colorisation d’image	20
I.4.1 Colorisation basée sur le gribouillis	21
I.4.2 Colorisation basée sur des exemplaires	22
I.4.3 Colorisation basée sur deep learning	23
I.4.4 Méthodes hybrides de colorisation d’images	24
I.5 Conclusion	25
II Le Deep learning	26
II.1 Introduction	27
II.2 Les Réseaux de neurones	27
II.3 Principe de Fonctionnement des réseaux de neurones	28
II.3.1 Convolution Neural Network (CNN)	29
II.3.2 Principe de la CNN	29
II.3.3 Les dimensions de la CNN	30
II.3.3.1 CNN 1 dimensionnel — Conv1D	30

II.3.3.2	CNN à 2 dimensions — Conv2D	30
II.3.3.3	CNN tridimensionnel — Conv3D	30
II.3.4	Les principaux composants du CNN	30
II.3.4.1	1D Convolutional layers	30
II.3.4.2	Pooling layers	31
II.3.4.3	Fully connected layers (FC)	32
II.3.4.4	Dropout layers	33
II.3.4.5	Loss function	33
II.3.5	Architecture	33
II.3.6	Couche CNN	34
II.3.6.1	Couche de convolution(CONV)	34
II.3.7	Couche pooling (POOL)	35
II.3.8	La couche de correction ReLU	36
II.3.9	Bonus : le padding	37
II.3.10	Paramètres des CNN	37
II.4	La colorisation des images medicales par deep learning	38
II.5	Conclusion	39
III	Colorisation des images médicales par CNN	40
III.1	Introduction	41
III.2	Environnement de développement	41
III.2.1	Google Colab	41
III.2.2	Pourquoi les GPU ?	42
III.2.3	Python	42
III.3	Description des bibliothèques utilisées	42
III.3.1	Pandas	42
III.3.2	NumPy	43
III.3.3	Matplotlib	43
III.3.4	OpenCV	44
III.3.5	TensorFlow	44
III.3.6	Keras	44
III.4	Architecture du système proposé	45

III.5 La base de données utilisées	46
III.5.1 Dataset	46
III.6 Description de SCIM	46
III.6.1 Définition des paramètres de SCIM	46
III.7 Préparation des données	48
III.7.1 Prétraitement des images	49
III.7.2 Données d'entraînement	49
III.7.3 Architecture de modèle proposé	50
III.7.4 Résultats du modèle de Colorisation	51
III.7.5 Prédiction	52
III.8 Conclusion	54
Conclusion	55

Table des figures

I.1	Processus de colorisation d'une image en niveaux de gris [1]	15
I.2	Ensemble des couleurs possibles pour des pixels de differents niveaux de gris [1]	16
I.3	Modèle de cube RVB normalisé [1]	17
I.4	Image RVB originale (499, 554 octets) [1]	17
I.5	Comparaison entre l'image couleur et l'image grise [1]	17
I.6	Exemple Colorisation basé sur le gribouillis [43]	21
I.7	Processus de colorisation par l'exemple [1]	23
I.8	Processus de colorisation basée sur deep learning [1]	24
II.1	Représentation du réseau de neurones humain [43].	27
II.2	Représentation du réseau de neurones artificiel [43].	28
II.3	Les fonctions D'activation [26]	31
II.4	Exemple sur le Maxpooling [26]	32
II.5	Représentation des fully-connected Layers [26]	32
II.6	Exemple sur le fonctionnement des Dropout Layers[26]	33
II.7	réseaux de neurone convolutif [26]	34
II.8	Exemple de filtre de taille (3*3)	35
II.9	Max poling [26]	36
II.10	Average pooling [26]	36
II.11	Allure de la fonction ReLU	36
II.12	Padding de 2 autour dune image de taille 32×32 [26]	37
II.13	Exemple de L'architecture de la méthode basée sur CNN pour colorer les images CT pulmonaires en niveaux de gris [9]	38
III.1	logo de colab	41

III.2 Le logo de Python	42
III.3 logo de numpy	43
III.4 logo de Matplotlib	43
III.5 logo de OpenCv	44
III.6 logo de TensorFlow	44
III.7 logo de keras	45
III.8 Les bibliothèques utilisées :	45
III.9 Les étapes de développement du model de colorisation.	45
III.10 la compilation du modèle.	47
III.11 Les paramètres d'entraînement du modèle.	47
III.12 Le modèle CNN.	47
III.13 Modèle de couleur de LAB	48
III.14 Normaliser et resize les images	49
III.15 convertir les images RGB en Lab.	50
III.16 Architecture du modèle CNN de colorisation	51
III.17 Courbes de précision et de perte sur la base d'apprentissage et la base de test	51
III.18 Évaluation du modèle de colorisation	52
III.19 Fonction de prédiction du modèle de Colorisation	52
III.20 les image obtenu par le model	53

Liste des tableaux

Introduction Générale

La coloration des images médicales est devenue un domaine de recherche de plus en plus important en imagerie médicale. Les images médicales sont souvent en niveaux de gris, ce qui peut rendre difficile l'interprétation et le diagnostic précis des maladies ou des anomalies par les cliniciens. En colorisant les images médicales, les chercheurs cherchent à fournir des informations visuelles supplémentaires pour aider au diagnostic, à la planification du traitement et au suivi des maladies.

Le processus de coloration des images médicales implique l'application de couleur aux images en niveaux de gris en utilisant diverses techniques de traitement d'image, telles que la segmentation, le filtrage et la fusion d'images. Il est également possible d'utiliser des méthodes basées sur l'apprentissage en profondeur (Deep learning) pour apprendre à prédire les couleurs des différentes structures anatomiques à partir des images médicales Dans ces dernières années, plusieurs études ont été réalisées, dans ce domain [45][46][47].

Cependant, il y a encore des défis dans la coloration des images médicales, tels que la préservation de l'intégrité des informations en niveaux de gris d'origine, l'assurance de la cohérence des couleurs et l'évitement de la mauvaise interprétation des informations colorées.

Dans ce projet, nous visons à automatiser la colorisation dans les visualisations d'imagerie médicale en exploitant les possibilités offertes par apprentissage profond (deep learning) Nous avons structuré et organisé notre manuscrit comme suit :

Chapitre 1 : "Colorisation des images".

Ce chapitre résume le concept, les domaines d'application et l'approche de la colorisation.

chapitre 2 : "Deep learning"

Dans ce chapitre, nous décrivons les réseaux de neurones convolutifs comme un modèle de deeplearning puis nous exposons l'utilisation de deeplearning dans le cas de colorisation des images médicales

Chapitre 3 : "Colorisation des images médicales par CNN"

Le but de ce chapitre est de présenter le processus de développement de notre système de colorisation des images médicales à base de CNN ,en présentant l logiciel et matériel sur lequel le système est construit, ainsi que les langages de programmation et les outils utilisés. Au cours de ce chapitre, nous présenterons les algorithmes et interfaces que nous avons développés et ainsi que les résultats obtenus.

Enfin, le manuscrit se termine par une conclusion qui résume le travail réalisé et les résultats obtenus en donnant les améliorations possibles et quelques perspectives.

Chapitre I

Colorisation des images

I.1 Introduction

La colorisation d'images en niveaux de gris est un sujet relativement nouveau dans le domaine du traitement d'images. Bien que des tentatives manuelles de colorisation de films en noir et blanc aient été faites dans les années 1980 et avant pour la colorisation d'images, la recherche sur la colorisation automatique n'a émergé que ces dernières années. Différentes techniques ont été décrites dans la littérature, et le nombre de chercheurs et de technologies dans ce domaine ne cesse de croître chaque jour.

Dans ce chapitre nous allons donner un aperçu sur le domaine de colorisation des images en présentant les approches utilisées et les domaines d'application.

I.2 Définition

La colorisation est le processus de transformation d'une image en niveaux de gris - qui est en réalité une image avec 256 nuances de gris - en une image colorée réaliste. Cette technique est couramment utilisée dans l'industrie cinématographique pour rendre les images d'archives plus attrayantes pour le grand public. Les couleurs doivent être choisies en fonction de l'objet à coloriser et de son environnement. Les algorithmes de colorisation utilisent l'image en niveaux de gris comme canal de luminance. La luminance correspond au flux lumineux émis ou réfléchi par chaque objet de la scène et capturé par la caméra. Dans le cas d'une image en couleur, la partie du signal vidéo correspondant à la couleur est appelée la chrominance. Cette caractéristique est représentée informatiquement par deux canaux supplémentaires (Figure I.1) [1].

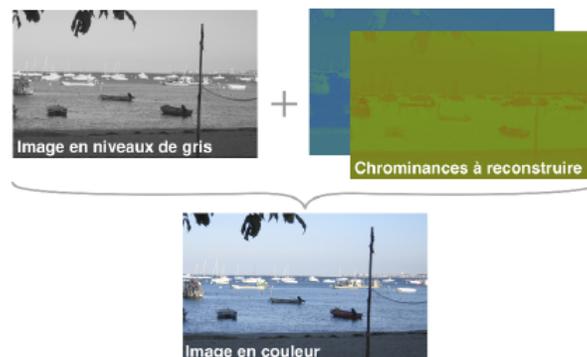


FIGURE I.1 – Processus de colorisation d'une image en niveaux de gris [1]

La colorisation d'une image consiste à choisir les valeurs de deux canaux de chrominance pour chaque pixel de l'image, après avoir converti celle-ci dans l'encodage standard Rouge, Vert, Bleu (RVB ou RGB en anglais) pour pouvoir l'afficher sur un écran. Si une couleur peut être représentée par un unique niveau de gris, la réciproque est fautive : à chaque niveau de gris correspond une palette spécifique de chrominances possibles, comme illustré dans Figure I.2. Pour lever cette ambiguïté, il existe plusieurs approches : demander à l'utilisateur de colorier partiellement l'image, fournir une image d'exemple à l'algorithme, ou utiliser des millions d'images pré-colorisées pour apprendre quelle couleur convient le mieux pour le contenu d'une image en noir et blanc [1].

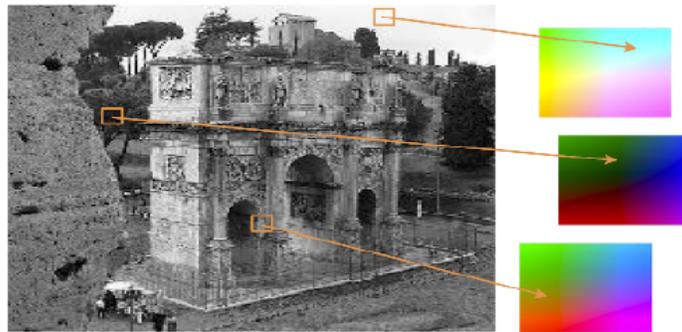


FIGURE I.2 – Ensemble des couleurs possibles pour des pixels de différents niveaux de gris [1]

Coloriser des images en niveaux de gris, également connu sous le nom de "colorisation", consiste à ajouter des couleurs à des images monochromatiques. Cela est devenu un domaine de recherche populaire car il améliore l'attrait visuel des images, en particulier des vieilles photos en noir et blanc, des films et des illustrations scientifiques, des images médicales. De plus, le contenu informationnel de certaines images scientifiques peut être amélioré en introduisant de la couleur pour exploiter les variations de chromaticité et de luminance.

Le processus de colorisation implique de définir la valeur de gris comme une équation des trois composantes primaires du modèle de couleur RVB, à savoir le rouge, le vert et le bleu. Il existe deux façons de décrire ce problème [22] :

- Intensité (le plus couramment utilisé) : $\text{Gray} : (\text{Red} + \text{Green} + \text{Blue}) / 3$
- Luminosité (norme NTSC) : $\text{Gray} : 0.299 \text{ Red} + 0.587 \text{ Green} + 0.114 \text{ Blue}$

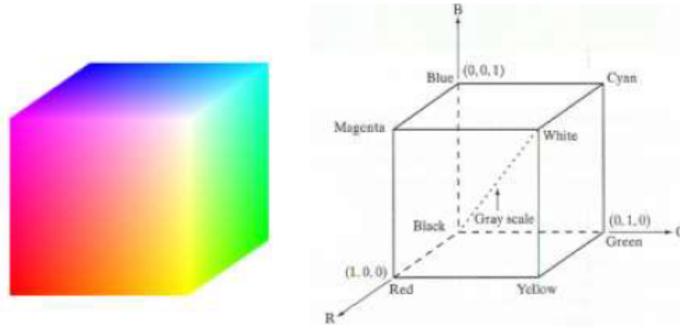


FIGURE I.3 – Modèle de cube RVB normalisé [1]



FIGURE I.4 – Image RVB originale (499, 554 octets) [1]

Les deux équations ne peuvent pas être inversées, ce qui implique qu'il est impossible de convertir n'importe quelle valeur de gris en ses composantes rouge, verte et bleue correspondantes. Étant donné qu'il existe $256 \times 256 \times 256$ couleurs possibles, il y a donc 256×256 combinaisons de couleurs totalement distinctes pour chaque valeur de gris allant de 0 à 255. Par exemple, la Figure I.5 présente diverses couleurs qui possèdent la même valeur de gris.

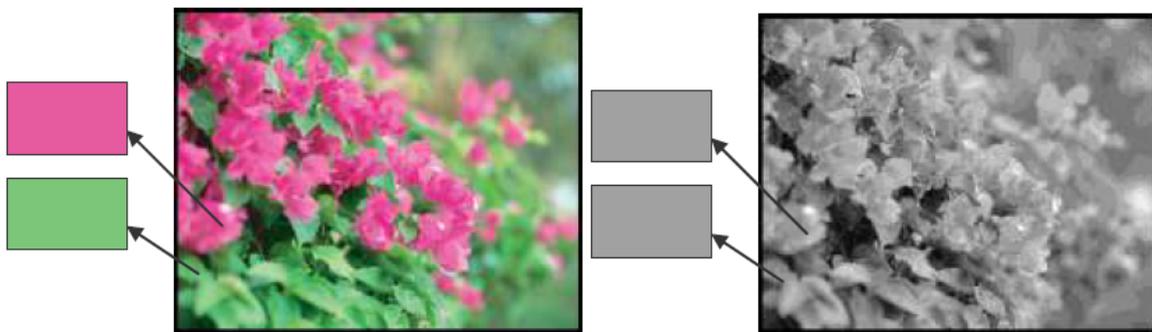


FIGURE I.5 – Comparaison entre l'image couleur et l'image gris [1]

I.3 Domaine d'application de colorisation

Outre l'amélioration des images délavées, la colorisation a suscité un intérêt considérable au cours des dernières décennies. De nombreux chercheurs se sont penchés sur le défi d'ajouter de la couleur aux films anciens en noir et blanc. Cette section examine certains de ces efforts, ainsi que d'autres domaines qui pourraient profiter de la colorisation d'images.

I.3.1 Colorisation des films :

Le procédé original a été développé par les Canadiens Wilson Markle et Brian Hunt [59] dans les années 1970. Il a été utilisé pour la première fois afin d'ajouter de la couleur aux photographies en noir et blanc de la mission Apollo.

En 1973, Carl Hansmann [14] a également déposé un brevet américain pour un système qui convertit les signaux monochromes en couleurs. Il a proposé l'intégration de ce circuit dans une caméra vidéo monochrome pour permettre l'enregistrement vidéo en couleur.

En 1987, Wilson Markle et son équipe [59] ont présenté leur méthode pour ajouter de la couleur aux films et émissions de télévision en noir et blanc. Leur algorithme de colorisation consiste à convertir un film en une bande vidéo et à dessiner manuellement un masque coloré pour au moins un cadre de référence dans un plan donné. Ensuite, ils utilisent la détection de mouvement et le suivi pour appliquer automatiquement les couleurs sur les autres images, dans les zones où il n'y a pas de mouvement. Cependant, lorsque les couleurs se rapprochent des bords en mouvement, le flux optique est utilisé, nécessitant parfois des ajustements manuels de la part de l'opérateur.

En 1988, Markle et son équipe [59] ont également déposé un brevet américain pour leur système de colorisation. Leur système convertit la bande vidéo en images fixes, puis un masque de couleur est dessiné manuellement pour au moins un cadre de référence dans chaque plan. Ensuite, ils utilisent la détection de mouvement et le suivi pour appliquer automatiquement les couleurs sur les images, dans les zones sans mouvement.

En 2004, le système Levine [15] a introduit une technique de colorisation utilisant des "graines" et le concept de flux lumineux. Cette approche a permis de colorer une série d'images.

En 2006, Takahiko Horiuchi et Hiroaki Kotera [16] ont proposé des algorithmes de co-

lorisation pour des séquences d'images monochromes de scènes fixes dans des clips vidéo. Leur méthode a obtenu des résultats de colorisation réussis en sélectionnant des images clés à intervalles réguliers de 10 images.

En 2009, Vivek George Jacob et Somana Gupta [17] ont introduit une méthode de colorisation semi-automatique dans laquelle l'utilisateur place manuellement des marqueurs colorés dans chaque zone pour indiquer la couleur souhaitée. L'algorithme segmente ensuite et colore l'image en fonction de la position et de la couleur de ces marqueurs. Pour la colorisation des clips vidéo, un petit ensemble d'images de référence est sélectionné manuellement à partir d'un ensemble plus large d'images clés générées automatiquement. Ces images sont colorisées en utilisant l'approche par balises, et les informations de couleur sont transférées vers les autres images de la vidéo en utilisant une technique de transfert de couleur basée sur l'estimation de mouvement.

I.3.2 Colorisation des images médicales

Limiter l'espace de l'image à la gamme de gris ne fournit que des différences de luminance et des textures en tant qu'indices visuels pour la reconnaissance d'objets dans l'image. Cependant, la couleur peut servir d'indice important et significatif pour détecter les formes et les objets. Une amélioration de la capacité de détection permet aux radiologistes d'interpréter les images en détail, améliorant ainsi la reconnaissance des objets et l'exactitude diagnostique. En moyenne, un observateur ne peut détecter que 140 niveaux de gris, alors qu'une image optimalement colorisée peut permettre à un utilisateur de distinguer entre 250 et 1000 niveaux différents, augmentant ainsi la détection potentielle des caractéristiques de l'image de 2 à 7 fois. La colorisation des images médicales peut être utilisée à diverses fins diagnostiques et servir d'outil de visualisation polyvalent [18].

En 2002, Paul J. Wojnicki et al. [18] ont mis en œuvre une carte de colorisation qui suit la carte de luminance des images en niveaux de gris d'origine. Le profil de luminance est préservé et la couleur est isolée comme mécanisme d'amélioration. L'effet de ce mécanisme d'amélioration sur la forme, la composition fréquentielle et les caractéristiques statistiques du potentiel évoqué visuel a été analysé et présenté.

Bochko et al. 2010 [19] ont présenté une méthode de colorisation pour les images médicales en niveaux de gris utilisant l'apprentissage des couleurs, où les couleurs sont apprises à partir

d'une image couleur et prédites pour une image en niveaux de gris. Leur algorithme utilise un modèle de réflexion dichromatique et a été étendu pour fonctionner efficacement avec des images en niveaux de gris qui ont un contraste plus faible que les images en couleur. Pour éviter un contraste négatif lors de la colorisation, ils ont proposé une technique fiable. Les chercheurs ont démontré la polyvalence de leur méthode avec deux applications médicales : l'étiquetage couleur réaliste des plaies cutanées et la colorisation des modèles de moulage dentaire.

En 2010, Jie Huang [20] a proposé une méthode de colorisation et un système de rehaussement d'image pour les images infrarouges et médicales. Ils ont utilisé la transformée en ondelettes, en tenant compte du système visuel humain, pour améliorer le contraste de l'image. La transformée en ondelettes a été utilisée pour décomposer les données de luminance en approximations et composants détaillés. Ils ont appliqué une technique d'amélioration du contraste basée sur le système visuel humain pour améliorer la composante approximative. Enfin, la transformée en ondelettes inverse a été appliquée pour obtenir une image couleur avec un contraste amélioré.

I.3.3 Compression d'image couleur

Les avancées rapides dans la qualité des techniques de colorisation ont incité les chercheurs à considérer l'idée de la décolorisation. La décolorisation consiste à convertir une image en couleur en une image en niveaux de gris tout en préservant certaines informations clés sur les couleurs d'origine qui seront utilisées dans le processus de recolorisation. En d'autres termes, elle vise à produire une image en niveaux de gris qui conserve des informations de couleur. Le concept derrière cette approche est de combiner les avantages de la colorisation et de la taille de fichier plus petite des images en niveaux de gris, ce qui donne une nouvelle image en couleur prenant moins d'espace que l'originale. Par conséquent, la décolorisation est considérée comme une méthode d'encodage des images en couleur [21]

I.4 Les approches de colorisation d'image

Dans ce paragraphe, nous voulons introduire les approche de colorisation. Il existe quatre approches pour explorer le problème de la colorisation d'image, y compris le gribouillis colorisation, colorisation basée sur des exemples, apprentissage automatique et apprentissage

profond colorisation et méthodes hybrides de colorisation.

I.4.1 Colorisation basée sur le gribouillis

La colorisation d'images basée sur le gribouillis est une tâche longue et coûteuse, généralement effectuée manuellement à l'aide d'Adobe Photoshop (PS) et d'autres logiciels. Par conséquent, on l'appelle aussi le gribouillis coloré marqué par l'homme. Une seule photo peut souvent prendre des semaines à colorier, par exemple le traitement d'un visage humain ne nécessite que 20 couches de nuances roses, vertes et bleues pour se recolorer correctement [2].

La colorisation basée sur le gribouillis est la première étape dans la recherche et la découverte de méthodes de colorisation pour les images et les vidéos en niveaux de gris. Cependant, le principal inconvénient de ces méthodes est que de ce processus de colorisation se fait entièrement manuellement. De plus, il faut beaucoup de travail acharné et de compétences professionnelles pour colorer une image avec une composition finement proportionnée afin d'obtenir un bon doodle. Dans le même temps, il dépense également des coûts de calcul supplémentaires pour obtenir des résultats de haute qualité [3].

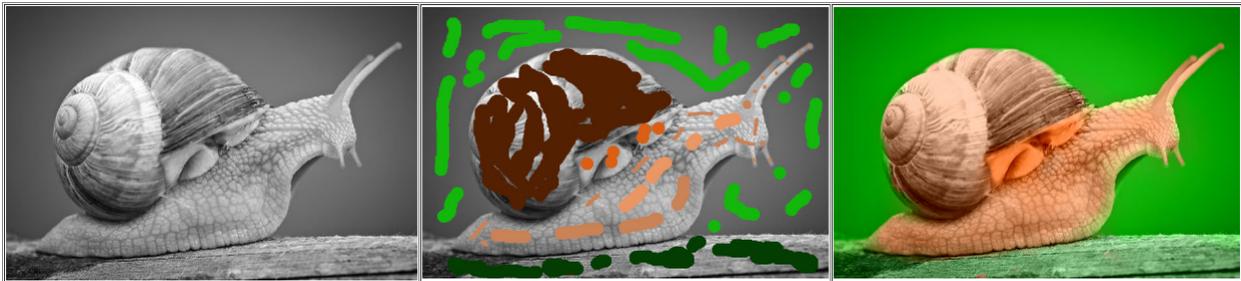


FIGURE I.6 – Exemple Colorisation basé sur le gribouillis [43]

Ces méthodes ont pour objectif de propager les indications locales de l'utilisateur, telles que les points ou les traits de couleur, dans l'ensemble de l'image en niveaux de gris. La propagation de la couleur est basée sur certaines métriques de similarité de bas niveau. Les travaux initiaux de Levin et al [15] ont supposé que les pixels adjacents ayant une luminance similaire devraient avoir des couleurs similaires, puis ont résolu un champ aléatoire de Markov pour propager les couleurs des griffonnages clairsemés. Les avancées ultérieures ont étendu la notion de similarité pour inclure les textures (Qu et al [55] , Luan et al [5]), la distance intrinsèque (Yatziv et Sapiro [58]) et ont utilisé les contours pour réduire les saignements de couleur (Huang et al [54]). La principale limitation de ces méthodes est la nécessité d'un

travail manuel intensif et d'une expertise professionnelle pour fournir des griffonnages de bonne qualité [23].

I.4.2 Colorisation basée sur des exemplaires

La sélection rigoureuse d'images couleur de référence capturées à partir de scènes présentant le même arrière-plan que l'image cible en niveaux de gris est une condition préalable essentielle pour la colorisation d'image basée sur des exemples. Cette approche est également appelée colorisation basée sur les patches. La première méthode repose sur une forme de correction des couleurs, utilisant une analyse statistique simple pour reconstruire une image cible à partir d'une image de référence [4]. Ces méthodes offrent une approche plus conviviale pour réduire la quantité d'effort nécessaire en fournissant une référence similaire à l'image en niveaux de gris en entrée. Les premières tentatives de transfert de couleurs, telles que Welsh et al et Reinhard et al. [4], se sont appuyées sur la correspondance des statistiques de couleur globale, mais cette approche a souvent donné des résultats insatisfaisants car elle négligeait les informations spatiales des pixels. Pour obtenir un transfert local plus précis, différentes techniques de correspondance ont été explorées, notamment au niveau des régions segmentées [Irony et al. 2005 [46] ; Tai et al [48], Charpiat et al [47]], au niveau des super-pixels , et au niveau des pixels.

Cependant, la recherche de correspondances de caractéristiques de bas niveau, telles que Gabor wavelet, à l'aide de métriques de similarité artisanales est vulnérable aux erreurs en cas de variation significative d'intensité et de contenu. Plus récemment, deux études ont utilisé des caractéristiques profondes extraites d'un réseau VGG-19 pré-entraîné pour une correspondance fiable entre des images qui sont sémantiquement liées mais visuellement différentes, et ont appliqué cela au transfert de style (Liao et al)[49] et au transfert de couleur (He et al)[23]. Cependant, toutes ces méthodes basées sur des exemples nécessitent toujours une bonne référence, ce qui peut être difficile pour les utilisateurs, même en utilisant des méthodes de récupération semi-automatiques [Liu et al [51], Chia et al[50]]. En revanche, cette approche est robuste à toute image de référence, grâce à la capacité du réseau profond à apprendre les distributions de couleur naturelles à partir de données d'image à grande échelle [23].

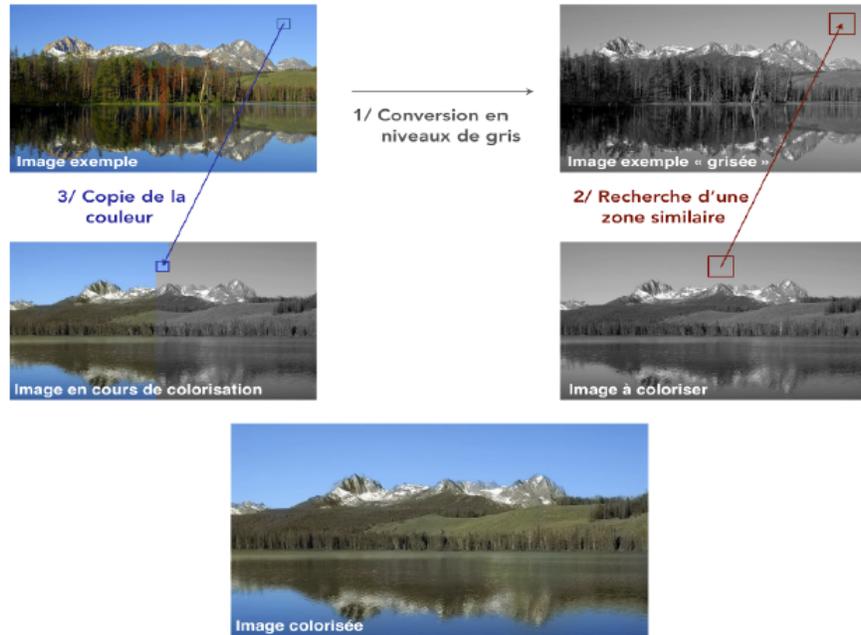


FIGURE I.7 – Processus de colorisation par l'exemple [1]

I.4.3 Colorisation basée sur deep learning

La méthode de colorisation repose sur la localisation des nombreux patches et pixels d'images similaires à partir de une image de référence massive, transférant les couleurs du bloc correspondant au pixel du image cible finale. Un résultat unique est produit pour chaque entrée [3]. La méthode de base de l'apprentissage en profondeur consiste à pour extraire automatiquement des caractéristiques à partir de données à grande échelle et prédire les couleurs possibles des images cibles. Les deux observations intuitives des informations sémantiques et de l'histogramme de couleur de chaque point de l'image se reflètent dans la conception. Basé sur l'excellente méthode d'appariement des patches, des données de référence à très grande échelle Pour la formation de modèles, y compris le filtrage bilatéral conjoint comme étape de post-traitement pour supprimer les artefacts.

Plusieurs techniques reposent entièrement sur l'apprentissage pour produire le résultat de colorisation. Deshpande et al [52] ont défini la colorisation comme un système linéaire et ont appris ses paramètres sont appris. Cheng et al [53] ont concaténé plusieurs fonctionnalités prédéfinies et les ont introduites dans un réseau de neurones entièrement connecté à trois couches. Récemment, certaines approches d'apprentissage de bout en bout [Larsson et al

[11], Iizuka et al [8], Zhang et al [6], Isola et al [57] ont exploité un réseau de neurones convolutifs pour extraire automatiquement des fonctionnalités et prédire le résultat de la couleur. La principale différence entre ces réseaux est la fonction de perte (par exemple, la perte de reconstruction d'image L2 (Iizuka et al[8]), la perte de classification (Larsson et al[54], Zhang et al[6]), et la perte L1 + GAN pour considérer la colorisation multimodale (Isola et al[60]). Tous ces réseaux sont appris à partir de données à grande échelle et ne nécessitent aucune intervention de l'utilisateur. Cependant, ils ne produisent qu'un seul résultat plausible pour chaque entrée, même si la colorisation est intrinsèquement un problème mal posé avec une incertitude multimodale (Charpiat et al[61]).

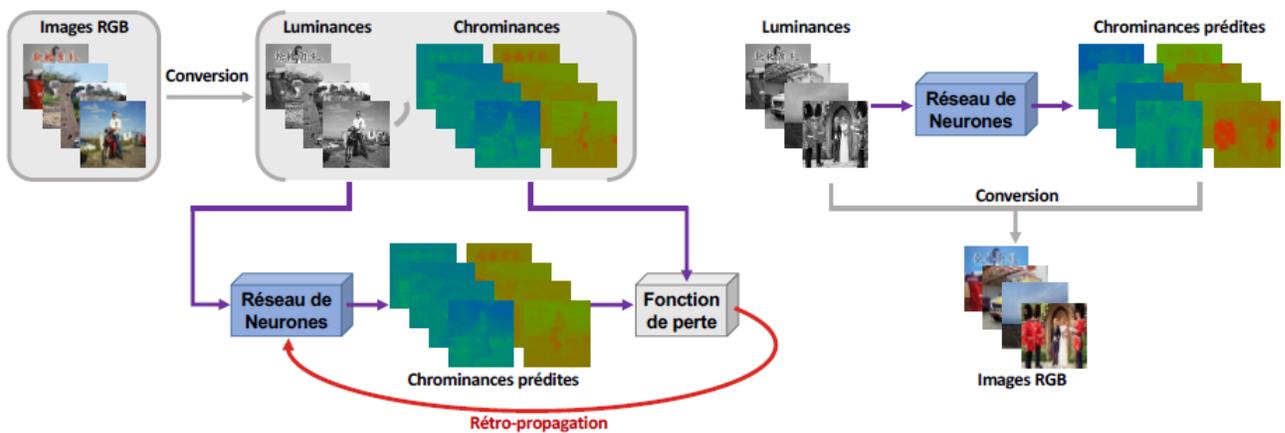


FIGURE I.8 – Processus de colorisation basée sur deep learning [1]

I.4.4 Méthodes hybrides de colorisation d'images

En 2017, Zhang et al.[6] ont proposé un modèle comparable, en utilisant une manière intuitive qui applique la récupération d'images pour compléter automatiquement la collection de référence. Des modèles de colorisation hybrides de CNN et Inception-ResNetV2 ont été employés par [6] en important les entrées dans le modèle. Ces modèles se concentrent sur les réseaux de neurones qui restituent des images en niveaux de gris à l'aide de 20 images couleur et atteignent un taux de satisfaction de 30 après le processus de colorisation de l'image

I.5 Conclusion

Dans ce chapitre, nous avons présenté le domaine de colorisation des images et ses applications. Nous avons un aperçu sur les travaux réalisés, dans ce domaine, avec les différentes approches utilisées.

Dans le chapitre suivant, nous allons explorer le domaine de deep learning et son utilisation dans la colorisation des images médicales.

Chapitre II

Le Deep learning

II.1 Introduction

Le deep learning ou apprentissage profond est un type d'intelligence artificielle dérivé de la machine learning (apprentissage automatique). Cette forme d'intelligence artificielle fonctionne avec des algorithmes capables d'imiter le cerveau humain grâce à un large réseau de neurones artificiels.

Dans ce chapitre, nous donnons le principe de réseaux de neurones artificiels (RNA) puis nous présentons l'architecture des réseaux de neurones convolutifs (Convolutional Neural Network ou CNN) et enfin nous exposons l'utilisation de deep learning dans la colorisation des images médicales

II.2 Les Réseaux de neurones

Le réseau de neurones est avant tout un concept, qui n'a pas de réalité physique. Il a été inspiré par les neurones biologiques (Figure II.1) et a donné naissance aux réseaux de neurones artificiels (ANN) (Figure II.2). Dans un réseau de neurones biologiques, plusieurs neurones travaillent ensemble pour recevoir des signaux d'entrée, traiter des informations et produire un signal de sortie. Les réseaux de neurones artificiels sont une forme de technologie de Deep Learning, qui fait elle-même partie de la sous-catégorie de l'Intelligence Artificielle appelée Machine Learning (apprentissage automatique)[43].



FIGURE II.1 – Représentation du réseau de neurones humain [43].

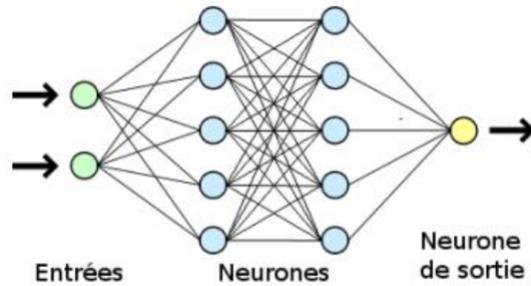


FIGURE II.2 – Représentation du réseau de neurones artificiel [43].

II.3 Principe de Fonctionnement des réseaux de neurones

De manière générale, un réseau de neurones est constitué de nombreux processeurs qui fonctionnent en parallèle et sont organisés en couches. La première couche reçoit les entrées d'informations brutes, de la même manière que les nerfs optiques de l'être humain lorsqu'il traite des signaux visuels. Ensuite, chaque couche reçoit les sorties d'informations de la couche précédente [25].

Le processus est similaire chez l'être humain, lorsque les neurones reçoivent des signaux en provenance des neurones proches du nerf optique. Finalement, la dernière couche produit les résultats du système. Les réseaux de neurones apprennent à travers un algorithme, qui permet à l'ordinateur d'apprendre à partir de nouvelles données. L'ordinateur équipé d'un réseau de neurones apprend à effectuer une tâche en analysant des exemples pour s'entraîner.

Ces exemples sont pré-marqués afin que le réseau puisse voir ce qu'ils sont. Par exemple, un réseau de neurones peut être utilisé pour apprendre à un ordinateur à colorier des images en niveaux de gris. Un grand nombre d'images en niveaux de gris et en couleurs correspondantes sont présentées au réseau neuronal, et l'ordinateur apprend à reconnaître la couleur de chaque pixel dans les nouvelles images en analysant des motifs répétitifs dans les images échantillons. Ainsi, en analysant des milliers d'images en couleurs, le réseau de neurones apprendra à colorer une image en niveaux de gris.

Les réseaux de neurones ne peuvent pas être programmés directement pour effectuer une tâche, contrairement à d'autres types d'algorithmes. Comme le cerveau en développement d'un enfant, la seule instruction qu'ils ont est d'apprendre. Cependant, il existe trois méthodes d'apprentissage distinctes. Dans le cas de l'apprentissage supervisé, l'algorithme s'entraîne

sur un ensemble de données étiquetées et se modifie jusqu'à être capable de traiter le dataset pour obtenir le résultat souhaité. Dans le cas de l'apprentissage non supervisé, les données ne sont pas étiquetées.

Le réseau de neurones analyse l'ensemble de données, et une fonction-coût lui indique dans quelle mesure il est éloigné du résultat souhaité. Le réseau s'adapte alors pour augmenter la précision de l'algorithme. Enfin, avec la méthode de l'apprentissage renforcé, le réseau de neurones est renforcé pour les résultats positifs et sanctionné pour les résultats négatifs, ce qui lui permet d'apprendre au fil du temps, de la même manière qu'un être humain apprend progressivement de ses erreurs. [25]

II.3.1 Convolution Neural Network (CNN)

Les réseaux de neurones convolutifs (CNN ou ConvNet - Convolutional Neural Network) sont un type spécial de réseaux de neurones multicouches à propagation avant souvent utilisés pour le traitement d'images. Ils ont été introduits initialement par Fukushima en 1980 pour la reconnaissance de formes, en s'inspirant de l'architecture du cortex visuel des animaux, et ont été popularisés pour la reconnaissance de caractères. De nos jours, les CNN sont largement utilisés dans de nombreuses applications d'intelligence artificielle, notamment le traitement automatique des langues, la classification de texte, la classification de musique et d'environnements sonores, ainsi que le traitement de signal.

II.3.2 Principe de la CNN

Les réseaux de neurones convolutionnels sont des réseaux neuronaux profonds qui comportent des couches de convolution ainsi que des couches entièrement connectées. La convolution est une opération mathématique couramment utilisée dans le traitement du signal. Dans les réseaux de neurones convolutionnels, les couches de convolution utilisent en fait la technique de corrélation croisée, qui est techniquement très similaire à la convolution. Il existe deux couches clés dans les réseaux neuronaux convolutionnels : les couches d'extraction de caractéristiques et les couches de classification [41].

II.3.3 Les dimensions de la CNN

Il existe 3 types d'opérations de convolution avec différentes dimensions 1D, 2D et 3D. La différence entre ces types de convolution se résume essentiellement au niveau des tailles des filtres, du nombre de directions de mouvement des filtres sur l'entrée A ainsi que de la dimension de la carte de caractéristiques produite en sortie [41].

II.3.3.1 CNN 1 dimensionnel — Conv1D

Elle est utilisée sur les séries temporelles, les données audio et textuelles car nous pouvons également représenter le son et les textes comme des données de séries temporelles [41].

II.3.3.2 CNN à 2 dimensions — Conv2D

Il s'agit du réseau neuronal à convolution standard qui a été introduit pour la première fois dans l'architecture Lenet-5. Conv2D est généralement utilisé sur des données d'image. Il est appelé CNN bidimensionnel parce que le noyau glisse le long de deux dimensions sur les données [41].

II.3.3.3 CNN tridimensionnel — Conv3D

Elle est utilisée avec des données d'images 3D. Comme les données d'imagerie par résonance magnétique (IRM). Les données IRM sont largement utilisées pour examiner le cerveau, la moelle épinière, les organes internes, etc [41].

II.3.4 Les principaux composants du CNN

Trois types de couches (en anglais Layers) composent le CNN : convolutional layers, pooling layers and fully-connected layers (FC). Lorsque ces couches sont empilées, une architecture CNN est formée. En plus de ces trois couches, il y a trois paramètres plus importants qui sont dropout layers, the activation function et loss function qui sont définis ci-dessous [20].

II.3.4.1 1D Convolutional layers

La couche de convolution est la première couche utilisée pour extraire différentes caractéristiques des données d'entrée. Elle effectue une opération mathématique de convolution entre les données d'entrée et un filtre de taille particulière $M \times M$. En faisant glisser le filtre sur

les données, un produit scalaire est calculé entre le filtre et les parties des données d'entrée correspondant à la taille du filtre (MxM)[26]

Chaque couche de convolution dans le modèle CNN est accompagnée d'une fonction d'activation, qui est l'un des paramètres les plus importants du modèle. Ces fonctions sont utilisées pour apprendre et approximer des relations continues et complexes entre les variables du réseau, et décident quelles informations du modèle doivent être transmises vers l'avant et lesquelles ne doivent pas l'être. Elles ajoutent une non-linéarité au réseau, ce qui lui permet de capturer des structures complexes dans les données. Plusieurs fonctions d'activation sont couramment utilisées, telles que ReLU, Softmax, Linear, TanH et Sigmoid(Figure II.3)[26].

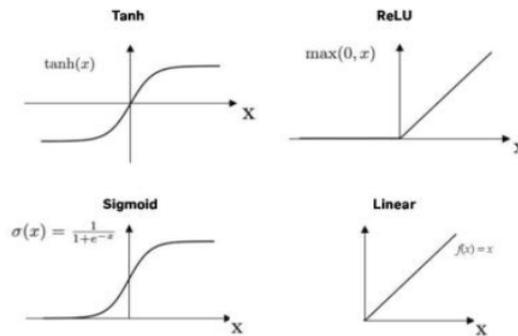


FIGURE II.3 – Les fonctions D'activation [26]

Chacune de ces fonctions a un usage spécifique en fonction du type de modèle CNN que l'on souhaite entraîner. Par exemple, pour un modèle de classification binaire, la fonction d'activation sigmoïde ou softmax est souvent utilisée, tandis que pour une classification multi-classes, la fonction softmax est plus courante. La sortie de la couche de convolution est appelée carte de caractéristiques, qui nous donne des informations sur les différentes caractéristiques présentes dans les données d'entrée. Cette carte de caractéristiques est ensuite introduite dans d'autres couches pour apprendre d'autres caractéristiques du Data d'entrée [26].

II.3.4.2 Pooling layers

La plupart des architectures CNN utilisent une couche de mise en commun (pooling layer) après chaque couche convolutive. L'objectif principal de cette couche est de réduire la taille de la carte de caractéristiques convolutionnelle afin de réduire les coûts de calcul et de rendre le réseau plus efficace. Pour ce faire, les opérations de pooling sont utilisées pour opérer

indépendamment sur chaque carte de caractéristiques et pour réduire les connexions entre les couches. Il existe plusieurs types d'opérations de pooling, mais la plus courante est le Maxpooling, qui extrait l'élément le plus grand de chaque section de la carte de caractéristiques. Le Pooling moyen (average pooling) calcule la moyenne des éléments de chaque section d'image, tandis que le Sum Pooling calcule la somme totale des éléments de chaque section [26].

La couche pooling agit comme un pont entre la couche de convolution et la couche entièrement connectée (FC). En réduisant la taille de la carte de caractéristiques, elle permet à la couche FC de mieux traiter les informations et de produire des prédictions plus précises [26].

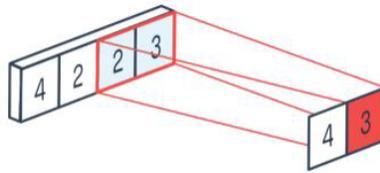


FIGURE II.4 – Exemple sur le Maxpooling [26]

II.3.4.3 Fully connected layers (FC)

La couche entièrement connectée (FC) est composée de poids, de biais et de neurones, et est utilisée pour connecter les neurones entre deux couches différentes. Elle est généralement placée avant la couche de sortie et constitue les dernières couches d'une architecture CNN (voir figure 12).

Dans cette couche, la carte de caractéristiques des couches précédentes est aplatie et transmise à la couche FC. Le vecteur aplati passe ensuite par plusieurs couches FC supplémentaires où des opérations mathématiques sont généralement effectuées. C'est à ce stade que le processus de classification commence à avoir lieu. [26]

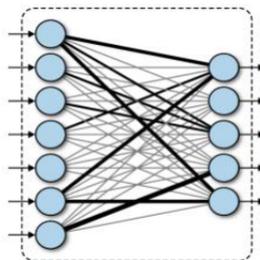


FIGURE II.5 – Représentation des fully-connected Layers [26]

II.3.4.4 Dropout layers

Lorsque toutes les caractéristiques sont reliées à la couche FC, cela peut souvent entraîner un surapprentissage dans l'ensemble de données d'apprentissage, où un modèle s'adapte si bien aux données d'apprentissage qu'il affecte négativement ses performances sur de nouvelles données [26].

Pour remédier à ce problème, une couche d'exclusion est utilisée pour éliminer certains neurones du réseau neuronal pendant le processus de formation, réduisant ainsi la taille du modèle. Par exemple, lorsque la couche d'exclusion est de 0,3, cela signifie que 30 des nœuds sont éliminés de manière aléatoire du réseau neuronal (Figure II.6).

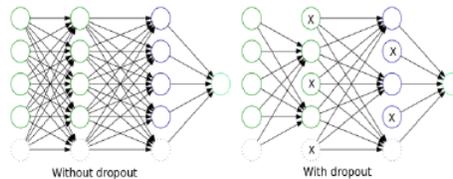


FIGURE II.6 – Exemple sur le fonctionnement des Dropout Layers[26]

II.3.4.5 Loss function

La dernière couche du réseau est appelée la couche de perte. Elle est responsable du calcul de l'erreur entre la prédiction du réseau et la valeur réelle. Dans le cas d'une tâche de classification, la variable aléatoire est discrète car elle ne peut prendre que les valeurs 0 ou 1, représentant respectivement la non-appartenance ou l'appartenance à une classe. C'est pour cette raison que la fonction de perte la plus courante et la plus appropriée est l'entropie croisée (cross-entropy) [26].

II.3.5 Architecture

Les réseaux de neurones à convolution sont fondés sur le perceptron multicouche (MLP) et s'inspirent du fonctionnement du cortex visuel des vertébrés. Bien que les MLP soient efficaces pour le traitement d'images, ils ont du mal à traiter des images de grande taille en raison de l'augmentation exponentielle du nombre de connexions avec la taille de l'image. Par conséquent, un réseau de neurones convolutif est composé de plusieurs couches, comme on peut le voir dans la figure II.7 [26].

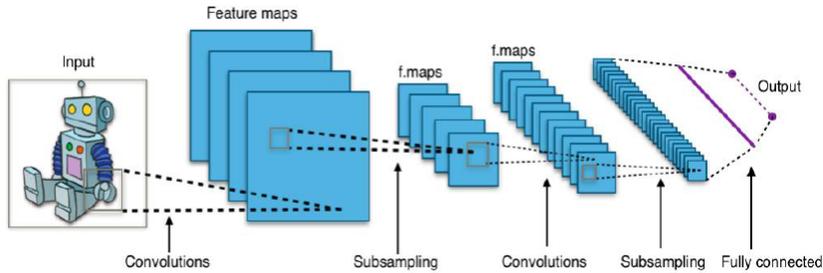


FIGURE II.7 – réseaux de neurone convolutif [26]

II.3.6 Couche CNN

Une architecture CNN est formée par un empilement de couches de traitement indépendantes [42] :

- La couche de convolution (CONV) qui traite les données d'un champ récepteur.
- La couche de pooling (POOL), qui permet de compresser l'information en réduisant la taille de l'image intermédiaire (souvent par sous-échantillonnage).
- La couche de correction (ReLU), souvent appelée par abus 'ReLU' en référence à la fonction d'activation (Unité de rectification linéaire).
- La couche "entièrement connectée" (FC), qui est une couche de type perceptron.
- La couche de perte (LOSS).

II.3.6.1 Couche de convolution(CONV)

Le terme "convolution" vient du fait que les réseaux CNN utilisent des opérations de convolution comme dans le traitement du signal ou dimages[14]. Dans la couche de convolution, on applique un produit de convolution entre l'image originale ou celle obtenue dans la couche précédente avec un filtre dont des coefficients sont assimilés à des poids synaptiques[26]. Le produit de convolution d'une matrice (image ou vecteur de mot dans ce cas) I avec un filtre K de taille $(s*s)$ est :

$$O(x, y) = \sum_{i=0}^{s-1} \sum_{j=0}^{s-1} k(i, j) \times I(x - 1 + j, y - 1 + i)$$

Il y a trois paramètres permettent de dimensionner le volume de la couche de convolution

la profondeur, le pas et la marge :

- Profondeur de la couche : Nombre de noyaux de convolution (ou nombre de neurones associés à un même champ récepteur).
- Le pas : Contrôle le chevauchement des champs récepteurs. Plus le pas est petit, plus les champs récepteurs se chevauchent et plus le volume de sortie sera grand.
- La marge (à 0) ou zero padding : Parfois, il est commode de mettre des zéros à la frontière du volume d'entrée. La taille de ce 'zero-padding' est le troisième hyper paramètre. Cette marge permet de contrôler la dimension spatiale du volume de sortie. En particulier, il est parfois souhaitable de conserver la même surface que celle du volume d'entrée. Cette marge permet de contrôler la dimension spatiale volume de sortie. En particulier.

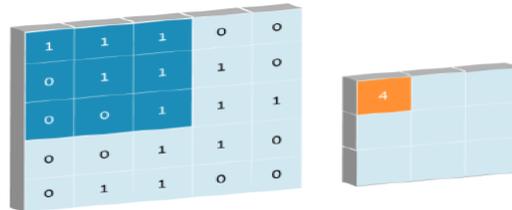


FIGURE II.8 – Exemple de filtre de taille (3*3)

II.3.7 Couche pooling (POOL)

La couche de pooling (ou couche de sous-échantillonnage en français) (POOL) est une opération qui est généralement appliquée après une couche de convolution dans un réseau de neurones convolutifs. Les deux types de pooling les plus couramment utilisés sont le max-pooling et l'average-pooling, où les valeurs maximales et moyennes sont sélectionnées, respectivement[26].

La couche de pooling permet de réduire le nombre de paramètres par conséquent d'accélérer les calculs dans le réseau. Elle permet aussi d'améliorer l'efficacité du réseau et éviter le problème sur-apprentissage[26].

- Max pooling : Chaque opération de pooling sélectionne la valeur maximale de la surface.

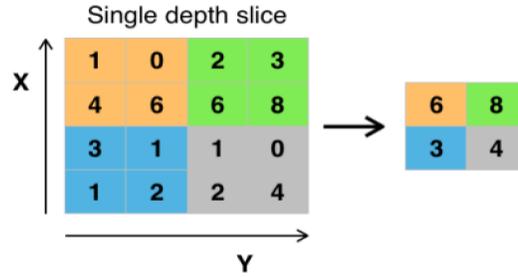


FIGURE II.9 – Max pooling [26]

— Average pooling : Il sous-échantillonne la carte des fonctionnalités, et Utilisé dans LeNet

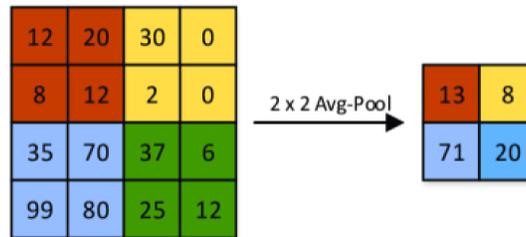


FIGURE II.10 – Average pooling [26]

II.3.8 La couche de correction ReLU

ReLU (Rectified Linear Units) désigne la fonction réelle non-linéaire définie par $\text{ReLU}(x) = \max(0, x)$. [19] La couche de correction ReLU remplace donc toutes les valeurs négatives reçues en entrées par des zéros. Elle joue le rôle de fonction d'activation. [26]

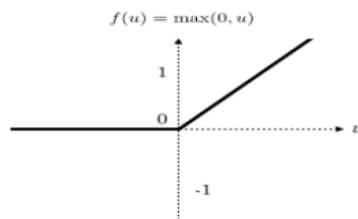


FIGURE II.11 – Allure de la fonction ReLU

À la fin de l'étape d'extraction d'informations, la dernière étape consiste en la mise à plat (flattening en anglais), qui consiste simplement à concaténer toutes les matrices pour en faire un seul vecteur (long). Les pixels (ou plutôt les éléments des matrices de nombres qui ne sont plus des images ou des pixels dans le cas des vecteurs de mots) sont récupérés ligne par ligne

et ajoutés au vecteur final.

II.3.9 Bonus : le padding

Le padding consiste à ajouter des zéros autour d'une matrice pour en augmenter la taille, afin de conserver une certaine dimension dans les tailles des images en sortie des convolutions. Parfois, cela peut être intéressant pour améliorer la qualité des prédictions ou pour mieux conserver les bords de l'image. Par exemple, un padding de 2 sur une matrice de taille 32×32 ajoutera des zéros à gauche sur 2 colonnes, à droite sur 2 colonnes, en haut sur 2 lignes et en bas sur 2 lignes [26].

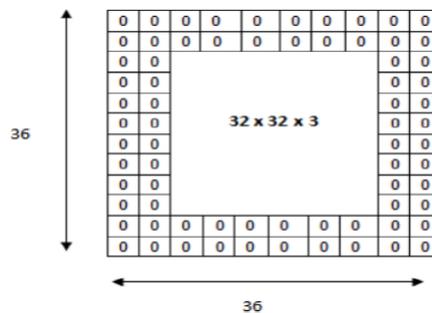


FIGURE II.12 – Padding de 2 autour d'une image de taille 32×32 [26]

Nous avons à présent tous les outils pour comprendre l'architecture d'un réseau de neurones convolutif. Il en existe plusieurs dans la littérature dont l'efficacité varie en fonction de la tâche, car ils n'ont pas tous le même nombre de convolutions (ni la même structure).

II.3.10 Paramètres des CNN

Dans les CNNs, le choix du nombre de couches de convolution, de correction ReLu, de pooling et de couches entièrement connectées est crucial. Il est également nécessaire de définir les paramètres pour chaque couche de convolution et de pooling. Pour dimensionner une couche de convolution, trois paramètres doivent être choisis : le nombre de noyaux de convolution, le pas de convolution (stride) et le zéro-padding (zero-padding) qui définit le nombre de zéros ajoutés sur les frontières des cartes, même s'il est parfois préférable de conserver la même taille que celle des cartes d'entrée. Quant à la couche de pooling, elle est définie par la taille de la fenêtre de traitement et le pas de chevauchement. En pratique, une fenêtre de (2×2) avec un pas de 1 est souvent choisie. [26]

II.4 La colorisation des images médicales par deep learning

La colorisation des images médicales par deep learning est un domaine de recherche en plein essor qui vise à ajouter des informations de couleur aux images médicales en niveaux de gris pour faciliter la visualisation et l'interprétation. L'utilisation du deep learning permet d'apprendre automatiquement les relations complexes entre les images en niveaux de gris et leurs correspondances en couleur. Les réseaux de neurones convolutifs (CNN) peuvent être utilisés pour la colorisation des images en niveaux de gris. L'idée de base derrière la colorisation avec les CNN est de former un modèle pour prédire la couleur de chaque pixel dans une image en niveaux de gris en fonction du contexte environnant. Pour colorer les images CT pulmonaires en niveaux de gris, Wang and Yan dans [9] ont proposé l'architecture de CNN présenté dans la Figure II.13. Le modèle proposé a été entraîné sur trois ensembles des images. Deux ensembles d'images avec un contenu complètement distinct sont utilisés pour explorer l'influence de l'ensemble de données sur le rendu des images CT pulmonaires. De plus un ensemble d'images de viande en couleur qui semble tout à fait adapté à la colorisation des images CT pulmonaires.

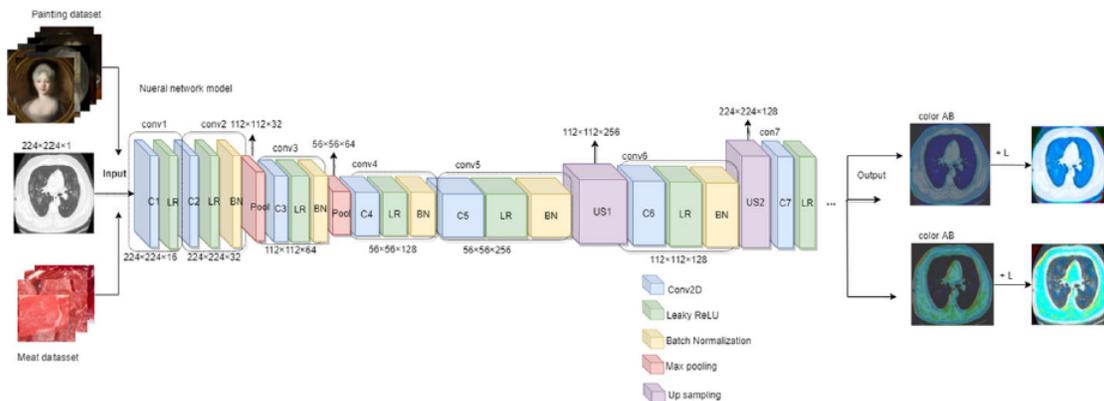


FIGURE II.13 – Exemple de L'architecture de la méthode basée sur CNN pour colorer les images CT pulmonaires en niveaux de gris [9]

Les réseaux antagonistes génératifs (GAN) sont également explorés pour la colorisation des images. Ils sont des modèles génératifs composés de deux parties opposées le générateur et la discrimination. Le travail du générateur est de produire une sortie qui est indiscernable de la réalité, tandis que le travail de la discrimination est de distinguer les images réelles des images générées. En plus de la perte de l'adversaire, la plupart des modèles utilisent également

une perte L1, ce qui force le générateur à produire des résultats qui sont structurellement similaires aux images de vérité terrain. Dans [56], Les auteurs ont proposé une architecture à base des réseaux antagonistes génératifs (GAN) pour la colorisation des images médicales et l'ont appliquée à des images en niveaux de gris provenant de différentes modalités, telles que l'imagerie par résonance magnétique (IRM) et la tomодensitométrie (TDM).

II.5 Conclusion

Dans ce chapitre, nous avons donné un tour d'horizon sur le paradigme de deep learning et son application dans le domaine de colorisation des images médicales. Nous avons détaillé la description du modèle CNN que nous allons l'exploiter dans développement de notre projet

Chapitre III

Colorisation des images médicales par CNN

III.1 Introduction

Dans ce chapitre nous décrivons le processus de développement de notre système à base CNN pour colorer les images médicales, en commençant par la présentation de l'environnement de développement avec les bibliothèques utilisés puis nous détaillons le principe de son fonctionnement.

III.2 Environnement de développement

III.2.1 Google Colab

Colab, développé par Google Research, est un environnement Jupyter portable gratuit qui fonctionne entièrement dans le cloud, éliminant ainsi le besoin de toute configuration. Il est spécialement conçu pour les techniques de Machine Learning et offre un service cloud gratuit, y compris un accès aux GPU. Cela permet aux utilisateurs d'améliorer leurs compétences en programmation Python[29][30].

Une caractéristique essentielle est les Notebooks créées par l'utilisateur peuvent être modifiés simultanément par d'autres membres de son choix afin d'assurer un meilleur travail collaboratif. Colab intègre toutes les bibliothèques essentielles pour le Machine Learning telles que PyTorch, TensorFlow, Keras, OpenCV, etc, , et permet d'importer ou enregistrer des Notebooks depuis ou vers Google Drive ainsi que la possibilité d'importer ou de publier des programmes ou des bases de données à partir de GitHub ou Kaggle[31].



FIGURE III.1 – logo de colab

III.2.2 Pourquoi les GPU ?

Les GPU (unités de traitement graphique) sont indispensables en raison du volume de données sur lequel les programmes vont opérer[67]. L'exécution de programmes de Deep Learning sur un CPU peut prendre des mois! Heureusement, en assignant ces tâches à des GPU, les calculs sont effectués beaucoup plus rapidement. Toutefois, les GPU sont souvent coûteux, c'est pourquoi Colab intervient en proposant gratuitement des GPU Nvidia Tesla K80. Ainsi, Colab facilite considérablement le travail des développeurs, en particulier dans les domaines de l'Intelligence Artificielle[29][31].

III.2.3 Python

Python est un langage de programmation puissant et facile à apprendre utilisé pour la programmation générale, mais efficace pour la programmation orientée objet. Il a été créé par Guido van Rossum et sorti en 1991[32][33][34]. L'utilisation des espaces dans Python est extrêmement importante, car elle permet des constructions qui favorisent une programmation claire, que ce soit à petite ou à grande échelle. Python offre une bibliothèque standard vaste et complète, et sa caractéristique la plus notable est sa prise en charge de multiples paradigmes de programmation, tels que l'orienté objet, l'impératif, le fonctionnel et le procédural. Les interpréteurs Python sont disponibles pour de nombreux systèmes d'exploitation[34].



FIGURE III.2 – Le logo de Python

III.3 Description des bibliothèques utilisées

III.3.1 Pandas

Est une bibliothèque open source sous licence qui fournit des structures de données et des outils d'analyse de données haute performance et faciles à utiliser pour le langage de programmation Python [35].

III.3.2 NumPy

est une bibliothèque conçue pour le calcul scientifique en Python. Elle fournit des objets de tableau multidimensionnel et des objets dérivés tels que des tableaux masqués et des matrices. NumPy propose un ensemble de routines permettant d'effectuer rapidement des opérations sur des tableaux, comprenant des fonctions mathématiques, logiques, de manipulation de formes, de tri, de sélection, de transformations discrètes de Fourier, d'algèbre linéaire de base, de statistiques élémentaires, de simulation aléatoire, et bien d'autres fonctionnalités encore[36].



FIGURE III.3 – logo de numpy

III.3.3 Matplotlib

Matplotlib est une bibliothèque de traçage complète pour créer des visualisations statiques, animées et interactives en Python, elle a beaucoup d'extension comme l'extension de mathématiques numériques NumPy[37].



FIGURE III.4 – logo de Matplotlib

III.3.4 OpenCV

OpenCV a été lancé chez Intel en 1999 par Gary Bradsky. La première version est sortie en 2000, il a été plus tard soutenu par Willow Garage puis Itseez [38].

OpenCV-Python est une bibliothèque de liaisons Python destinées à résoudre les problèmes de vision par ordinateur en temps réel [38].



FIGURE III.5 – logo de OpenCv

III.3.5 TensorFlow

TensorFlow est une plate-forme Open Source de bout en bout dédiée aux machines learning, créée et est maintenu par l'équipe Google Brain au sein de la machine de Google, une organisation de recherche en intelligence. TensorFlow publié sous la licence open source Apache 2.0 [39].



FIGURE III.6 – logo de TensorFlow

III.3.6 Keras

Keras est une API d'apprentissage profond écrite en Python, fonctionnant sur la plate-forme d'apprentissage automatique TensorFlow. Le but de cette bibliothèque est de permettre

la constitution rapide de réseaux neuronaux [40].



FIGURE III.7 – logo de keras

```
from keras.layers import Conv2D, UpSampling2D, InputLayer, Input
from keras.models import Sequential, Model
from keras.preprocessing.image import ImageDataGenerator
from skimage.io import imshow
from skimage.color import rgb2lab, lab2rgb, gray2rgb
from skimage.transform import resize
import skimage.io
from skimage.io import imsave
import numpy as np
import matplotlib.pyplot as plt
import tensorflow as tf
import keras
import os
import glob
import seaborn as sns
import pandas as pd
import cv2
```

FIGURE III.8 – Les bibliothèques utilisées :

III.4 Architecture du système proposé

Notre système de colorisation d'images médicales (SCIM) a entraîné sur une base d'apprentissage pour produire un modèle capable de coloriser les images médicales.

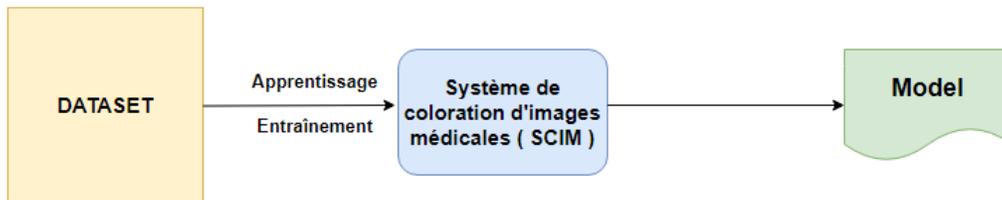


FIGURE III.9 – Les étapes de développement du model de colorisation.

III.5 La base de données utilisées

III.5.1 Dataset

Il s'agit d'une base de données qui contient 2024 images de tomographie du corps humain et qui se compose de deux catégories principales pour les mêmes images, les images en couleur et les images en niveaux de gris.

C'est une base de données qui contient 2024 coupes transversales pour les images CT et anatomiques, composée de deux classes principales une classe nommée "color" pour les images CT des image en color et la seconde nommée "gray" pour les meme images CT en niveau de gris.

Cette base de données a été créée par le NAME Visible Human Project®.

La base de données utilisé contient 2024 images. Nous l'avons divisé en 1416 (70%) image pour la entraînement, 404 (20%) image pour la validation ou pour l'évaluation du modèle et 204 (10%) image pour les tests.

III.6 Description de SCIM

La developpement de notre syst'eme SCIM necessite la description des points suivants :

III.6.1 Definition les param'etres de SCIM

Pour developper notre modèle d'apprentissage nous avons utilis'es les paramètres suivants :

- La taille des images : la taille originale des images est de (2048x1216), nous les avons redimensionnées à (256x256)
- La méthode de compilation : pour configurer le modèle d'entraînement, nous avons utiliser la méthode de compilation 'compile'
- L'optimiseur 'Adam' : il s'agit de l'un des optimiseurs les plus utilisés à cause de son rapidité. Il converge également de facon tres rapide pour corriger la latence du taux d'apprentissage et le contraste élevé.
- La fonction de perte (loss function) : 'mse' c'est un choix populaire de fonction de perte.
- Activation : pour garantir une meilleure performance de calcul nous utilisons dans tous les

couches 'ReLU', sauf la dernière 'tanh'.

- Metrics : est une fonction utilisée pour évaluer les performances du modèle, nous utilisons 'Accuracy'.

```
[ ] CNN.compile(loss='mse', optimizer='Adam', metrics=['accuracy'])
      CNN.summary()
```

FIGURE III.10 – la compilation du modèle.

- Epochs : les epochs sont le nombre de répétitions typiques de l'entraînement, dans notre programme le nombre des epochs est non déterminé, il est lié à la fonction d'arrêt, nous utilisant 150.

- Batch size : est un paramètre de l'algorithme d'apprentissage, représente le nombre des exemples de formation utilisés dans l'estimation du gradient d'erreur, nous utilisons 50.

```
CNN_model.fit(X_train = X,
              Y_train = Y,
              validation_split=0.2,
              epochs=150,
              batch_size=50)
```

FIGURE III.11 – Les paramètres d'entraînement du modèle.

D'après la Figure III.12, nous pouvons dire qu'il a plus de 3 millions de paramètres de formation. Afin d'entraîner le modèle, nous avons utilisé la fonction `model.fit()` du package Keras, Cette fonction nous permet d'adapter littéralement nos données aux maquette.

```
conv2d_11 (Conv2D)          (None, 128, 128, 16)    4624
conv2d_12 (Conv2D)          (None, 128, 128, 2)     290
up_sampling2d_2 (UpSampling (None, 256, 256, 2)    0
2D)
=====
Total params: 3,896,530
Trainable params: 3,896,530
Non-trainable params: 0
=====
```

FIGURE III.12 – Le modèle CNN.

III.7 Préparation des données

Dans cette étape les image dans RGB en converti a espace LAB.

L'espace colorimétrique Lab est une représentation numérique de la perception visuelle humaine, capable de décrire toutes les couleurs visibles par l'œil humain. Dans cet espace colorimétrique, les canaux de luminance et de couleur sont séparés. Le modèle de couleur Lab, illustré à la Figure III.13, présente le canal de luminance représenté par L, avec des valeurs allant de 0 à 100, indiquant la plage de luminance du noir au blanc. Les canaux de couleur a et b représentent l'espace de changement de couleur du rouge au vert et du jaune au bleu, respectivement. La plage de valeurs pour les deux canaux est $[-128, 127]$ [28].

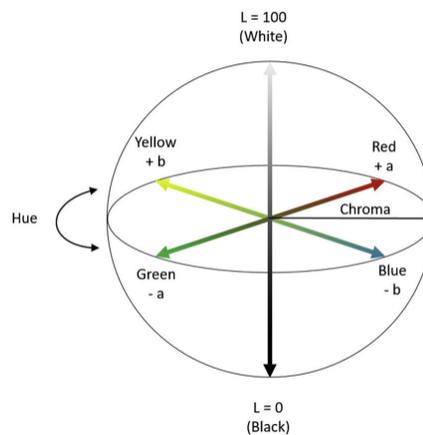


FIGURE III.13 – Modèle de couleur de LAB

L'espace de couleur Lab est reconnu comme un espace de couleur indépendant de l'appareil qui décrit la façon dont les couleurs sont affichées, plutôt que le nombre de couleurs qu'un appareil doit générer pour produire une couleur spécifique. Sa large gamme de couleurs, sa riche palette de couleurs et sa séparation de la luminance et de la chromaticité en font un outil précieux utilisé dans divers domaines. Par exemple, l'espace de couleur Lab est une application typique dans Photoshop, et il est couramment utilisé comme modèle dans les algorithmes de traitement d'image liés à la couleur. L'objectif de la colorisation d'images est de développer une fonction de mappage d'images à un canal vers des images à trois canaux, où l'espace colorimétrique Lab présente des avantages particuliers. Dans cet espace colorimétrique, le canal de luminance (L) peut être considéré comme une image en niveaux de gris, qui peut être directement utilisée pour l'entraînement de réseaux de neurones. La

fonction de mappage doit simplement générer deux canaux de couleur (a et b), tout en utilisant le canal de luminance d'origine pour créer une image en couleur. Cette approche préserve la plupart des caractéristiques de l'image d'entrée dans la sortie et maintient une similarité visuelle entre l'entrée et la sortie. De plus, la génération de seulement deux canaux simplifie la tâche et facilite l'entraînement du réseau de neurones [28].

III.7.1 Prétraitement des images

Nous utilisons la classe ImageDataGenerator pour créer des générateurs d'images. Cela nous permet de lire et de décoder des images, les convertissant ainsi en tenseurs à virgule flottante. Nous utilisons l'argument 'rescale' pour mettre à l'échelle et normaliser les valeurs du tenseur de la plage 0-255 à la plage 0-1.

Pour cela, nous faisons appel aux méthode de la classe ImageDataGenerator, notamment la méthode "flow from directory". Cette méthode s'avère particulièrement pratique lorsque les images sont organisées et placées dans leurs dossiers respectifs correspondant à leurs classes. En utilisant cette méthode, le système identifie automatiquement les classes à partir des noms des dossiers.

```
path = '/content/drive/MyDrive/Dataset/train'

#Normaliser les images - diviser par 255
train_datagen = ImageDataGenerator(rescale=1. / 255)

#Resize images, if needed
train = train_datagen.flow_from_directory(path,
                                         batch_size = 340,
                                         target_size=(256, 256),
                                         class_mode=None)

Found 1416 images belonging to 2 classes.
```

FIGURE III.14 – Normaliser et resize les images

III.7.2 Données d'entraînement

Nous convertissons les images RVB (RGB en anglais)en des images de LAB et séparons les canaux «L», «A» et «B» de l'image pour former le modèle pour prédire les valeurs de

«A» et «B».

```
#itérer sur chaque image et convertir le RVB en Lab.
X = []
Y = []
for img in train[0]:
    try:
        lab = rgb2lab(img)
        X.append(lab[:, :, 0])
        Y.append(lab[:, :, 1:] / 128) #Les valeurs A et B vont de -127 à 128,
        #nous divisons donc les valeurs par 128 pour limiter les valeurs entre -1 et 1.
    except:
        print('error')
```

FIGURE III.15 – convertir les images RGB en Lab.

III.7.3 Architecture de modèle proposé

La structure de notre modèle de colorisation d’image basé sur CNN est présentée dans la Fig Figure III.16

Encodeur : l’image d’entrée est redimensionnée à 256×256 . L’entrée représentée par $H \times L \times 1$ (composante L) est une image en niveaux de gris. Tandis que la sortie $H/8 \times W/8 \times 512$ comporte une réperestation. Il utilise 13 couches convolutionnelles avec un filtre 3×3 qui alterne la foulée 1 et le rembourrage pour préserver la taille d’entrée, la foulée 2 pour réduire de moitié la taille d’entrée. Le réseau de codeurs, chaque couche convolutive utilise une fonction d’activation ReLu.

Décodeur : Pour créer des couches convolutionnelles à sur-échantillonnage. La sortie finale $H \times W \times 2$ (composant ab) qui applique une série de couches convolutionnelles. Pour la dernière couche, nous utilisons tanh au lieu de Relu. En effet, nous colorons l’image dans ce calque à l’aide de 2 filtres, A et B. Les valeurs A et B sont comprises entre -1 et 1, de sorte que tanh (ou tangente hyperbolique) est utilisée car elle est également comprise entre -1 et 1. Les autres fonctions vont de 0 à 1.

Deux procédures nécessaires sont nécessaires pour traiter les images originales (données d’apprentissage) pour le rendu des images en niveaux de gris en couleurs.

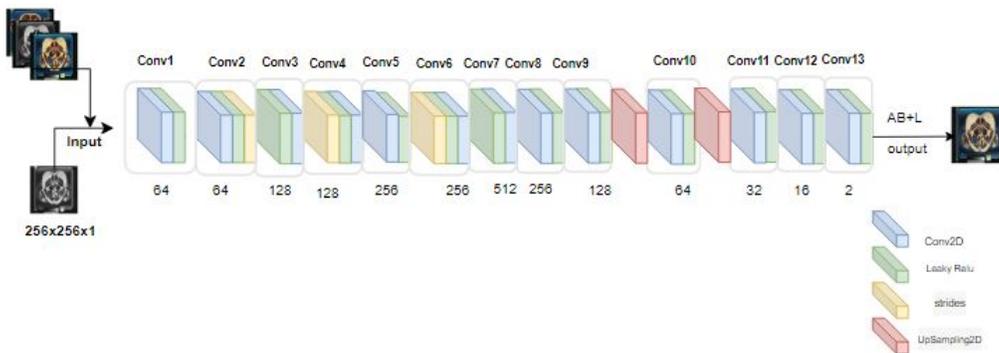


FIGURE III.16 – Architecture du modèle CNN de colorisation

III.7.4 Résultats du modèle de Colorisation

Après l'utilisation des données précédentes, les courbes obtenues dans les deux figures a et b, Celles-ci démontrent la stabilité des valeurs des fonctions du loss pendant la période d'entraînement, tandis que les fonctions de perte ont atteint leur niveau le plus bas. Nous avons également procédé à une vérification manuelle des images prédites (générées) une fois la formation du modèle terminée. Nous avons constaté que lorsqu'un modèle était formé pendant 150 époques, les images couleur présentaient des artefacts étranges, ce qui indique une sur-adaptation du modèle. Par conséquent, nous avons décidé d'adopter les paramètres mentionnés dans la première phrase précédente et de les appliquer à toutes les tentatives effectuées sur l'ensemble de données DATASET.

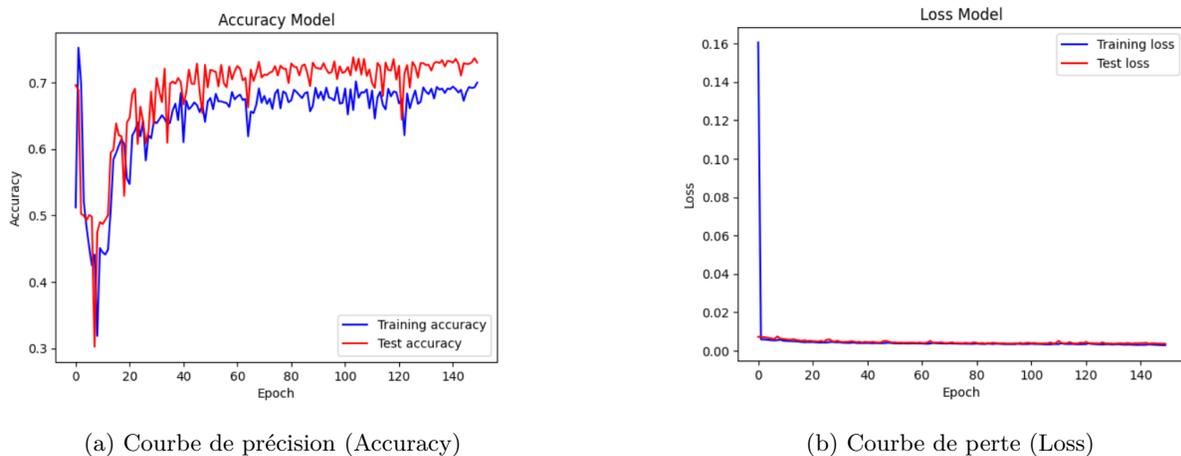


FIGURE III.17 – Courbes de précision et de perte sur la base d'apprentissage et la base de test

Le modèle a terminé l'entraînement avec un taux de précision d'environ 70%, et d'une diminution dans le loss 0.03%. La Figure III.18 montre l'évaluation du modèle.

```
model = tf.keras.models.load_model('/content/drive/MyDrive/Dataset/Model/Col2.model')
model.evaluate(X_val,Y_val)

11/11 [=====] - 7s 330ms/step - loss: 0.0037 - accuracy: 0.7014
[0.0036591971293091774, 0.7013532519340515]
```

FIGURE III.18 – Évaluation du modèle de colorisation

III.7.5 Prédiction

La prédiction est effectuée sur les images de la partie Test de la base. Premièrement, nous testons notre modèle sur une image, l'image obtenu par le système. Puis nous testons le

```
from keras.utils.image_dataset import load_image
from keras.utils import img_to_array
tf.keras.models.load_model(
    '/content/drive/MyDrive/Dataset/Model/Col2.model',
    custom_objects=None,
    compile=True)
img1_color=[]
img1=tf.keras.utils.img_to_array (tf.keras.utils.load_img('/content/drive/MyDrive/Dataset/test/Gray/image687.png'))
img1 = resize(img1 ,(256,256))
img1_color.append(img1)
img1_color = np.array(img1_color, dtype=float)
img1_color = rgb2lab(1.0/255*img1_color)[:,:,:0]
img1_color = img1_color.reshape(img1_color.shape+(1,))
output1 = CNN_model.predict(img1_color)
output1 = output1*128
result = np.zeros((256, 256, 3))
result[:,:,:0] = img1_color[0][:,:,:0]
result[:, :,1:] = output1[0]
imshow(lab2rgb(result))
```

FIGURE III.19 – Fonction de prédiction du modèle de Colorisation

modèle sur la partie Test de la base nous présentons quelques résultats qui sont exprimés dans la Figure III.20.

Nous testons le modèle sur la partie Test de la base nous présentons quelques résultats dans les figures suivantes :

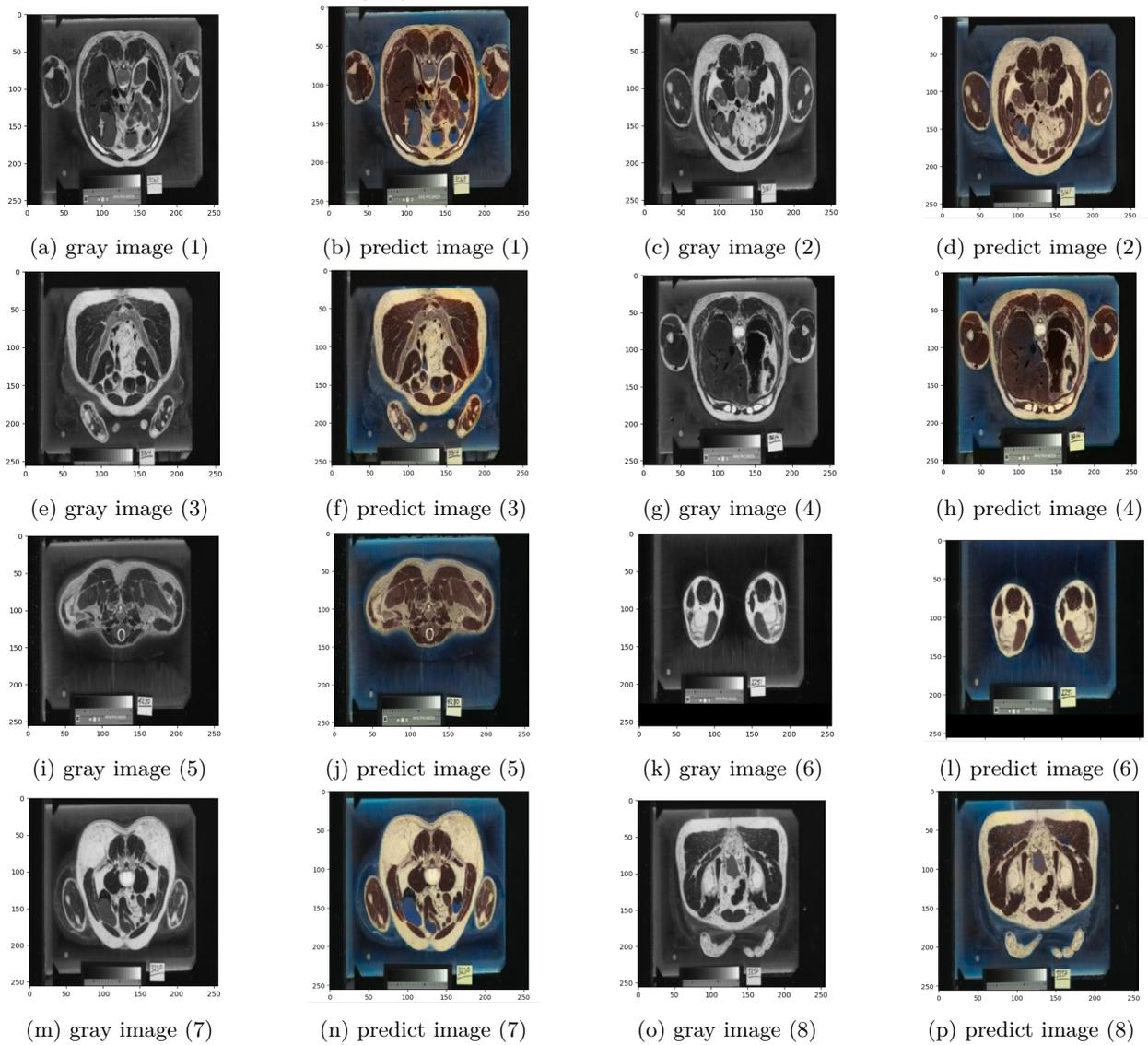


FIGURE III.20 – les images obtenues par le modèle

III.8 Conclusion

Nous avons présenté dans ce chapitre les outils utilisés pour l'implémentation de modèle proposé. Nous avons également présenté quelques résultats qualitatifs et quantitatifs des modèles proposés sur un ensemble des images CT.

Les résultats montrent que le deep learning permet de combler le manque de données pour aboutir à une performance acceptable par rapport à une base riche de données. Les résultats montrent aussi que le modèle CNN donne de bons résultats de Colorisation des images CT .

Conclusion générale

Nous avons abordé dans ce projet, le problème de colorisation des images médicales qui est un sujet d'actualité, vu que les images médicales, produites par tous les modalités, sont généralement en niveaux de gris et la colorisation de ces images permet d'ajouter des informations supplémentaires et de faciliter la compréhension des structures anatomiques et des anomalies présentes.

L'utilisation de modèles de deep learning, tels que les réseaux de neurones convolutifs, nous a permis d'obtenir des résultats prometteurs en termes de précision et de qualité de la colorisation. Ces modèles ont été entraînés sur de vastes ensembles de données et ont appris à extraire des caractéristiques pertinentes pour la colorisation des images médicales.

Cependant, malgré l'importance de résultats obtenus, il reste encore des défis à relever. L'une des limitations est la disponibilité de grandes bases de données d'images médicales annotées pour l'entraînement des modèles. De plus, la précision de la colorisation peut varier en fonction des caractéristiques spécifiques de chaque image et de la complexité des structures anatomiques à coloriser.

Néanmoins, cette étude ouvre la voie à de nombreuses applications potentielles dans le domaine médical. La colorisation des images médicales peut faciliter le travail des professionnels de la santé en fournissant des informations visuelles supplémentaires, en améliorant le diagnostic et en facilitant la communication avec les patients.

En conclusion, la colorisation des images médicales par le biais de l'apprentissage en profondeur présente un grand potentiel pour améliorer la visualisation et l'interprétation des images médicales, et offre de nouvelles perspectives pour la recherche future dans ce domaine.

Bibliographie

- [1] Aurélie Bugeau, Nicolas Papadakis, Apprendre à coloriser des images, 23 Jul 2019.
- [2] Abhishek Thakur et al, Image Colorization Using Deep Learning, International Journal of Scientific Research Engineering Trends, Nov-Dec-2019.
- [3] Yuewei Wang et Wei Qi Yan, Colorizing Grayscale CT images of human lungs using deep learning methods, Multimedia Tools and Applications, (2022)
- [4] E. Reinhard et al, Color transfer between images, IEEE Computer Graphics and Applications, (2001) .
- [5] Q. Luan et al, Natural image colorization. In Proc. of the 18th Eurographics conference on Rendering Techniques, Eurographics Association, (2005).
- [6] Richard Zhang et al, Real-time user-guided image colorization with learned deep priors. ACM Trans. Graph. (Proc. of SIGGRAPH) (2017)
- [7] Khan, M.U.G et al, Medical image colorization for better visualization and segmentation, Edinburgh. Communications in Computer and Information Science,, (2017)
- [8] Satoshi Iizuka et al, Let there be color! : joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification. ACM Trans. Graph. (Proc. of SIGGRAPH), (2016)
- [9] Elisa Mariarosaria Farella et al, Colorizing the Past : Deep Learning for the Automatic Colorization of Historical Aerial Images, Journal of Imaging, October 2022.
- [10] R. Zhang et al, Colorful Image Colorization, In Proceedings of the European Conference on Computer Vision, 2016.
- [11] G. Larsson et al, Learning representations for automatic colorization, In Proceedings of the European Conference on Computer Vision, 2016.

- [12] Apurva Pavaskar et al, COLORIZATION OF GRAYSCALE IMAGES USING DEEP LEARNING, International Journal of Computer Science and Engineering, 2020.
- [13] KOLEINI, M et al, Film colorization using texture feature coding and artificial neural networks, in Journal Of Multimedia. (2009).
- [14] HANSEMAN, C. Video monochrome to color conversion, 1973.
- [15] A. LEVIN et al, Colorization using optimization, SIGGRAPH in ACM Transactions on Graphics, July 2004.
- [16] T. HORIUCHI et al, Colorization Algorithm for Monochrome Video by Sowing Color Seeds, Journal of Imaging Science and Technology, 2006.
- [17] V. JACOB et al, Colorization of grayscale images and videos using a semiautomatic approach, In 16th IEEE International Conference on Image Processing, 2009.
- [18] P. WOJNICKI, Quantifying the effect of colorization enhancement on mammogram images, The International Society for Optical Engineering, 2002.
- [19] V. BOCHKO et al., Medical image colorization using learning, International Symposium on Multispectral Colour Science (CGIV 2010).
- [20] J. HUANG, Enhancement and colorization of infrared and other medical images, these Master, Department of Electrical and Computer Engineering, Lehigh University, 2010.
- [21] Noura Abd El-Moez Semary, IMAGE COLORING TECHNIQUES AND APPLICATIONS, Thesis Doctor, The University of Menofia Faculty of Computers and Information Department of Information Technology, 2011.
- [22] R. GONZALEZ et al, Digital Image Processing Using Matlab, 2004.
- [23] Mingming He et al, Deep Exemplar-based Colorization, Hong Kong UST, University of Science and Technology of China, 2018
- [24] H. Lu et Q. Zhang, Applications of Deep Convolutional Neural Network in Computer Vision, Engineering Intelligent Systems, 2016.
- [25] [https ://www.lebigdata.fr/reseau-de-neurones-artificiels-definition](https://www.lebigdata.fr/reseau-de-neurones-artificiels-definition) (accédé Le 18 Mai 2023).
- [26] Saadi Khaled Iben El Walid et Bendahi Khaled, Segmentation D'Image Médicale Via Non supervisor Réseau de neurones convolutif , thèse MASTER ACADEMIQUE,

Domaine : Electronique, Option : Electronique des Systèmes, Faculté Technologie, Département de Electronique, UNIVERSITE MOHAMED BOUDIAF - M'SILA , 2021/2022.

- [27] V M Spitzer, D G Whitlock, The Visible Human Dataset : the anatomical platform for human simulation, 1998.
- [28] TAO DENG, Image Colorization Based on Deep learning, Master's Programme, Machine Learning, School of Electrical Engineering and Computer Science Swedish, 2023.
- [29] G. RESEARCH, <https://research.google.com/colaboratory/faq.html> : : text=Colaboratory, (accédé Le 18 Mai 2023).
- [30] TUTORIALSPPOINT, https://www.tutorialspoint.com/google_colab/what_is_google_colab.htm : :text: (accédé Le 18 Mai 2023).
- [31] MEDIUM, <https://medium.com/deep-learning-turkey/google-colab-freegpu-tutorial-e113627b9f5d>, (accédé Le 19 Mai 2023).
- [32] G. VAN ROSSUM et F. L. DRAKE, An introduction to Python, Livre Bristol, 2003.
- [33] G. VAN ROSSUM et F. L. DRAKE JR, Python Reference Manual. Python tutorial, Livre, 1995.
- [34] PYTHON, <https://www.python.org/>, (accédé Le 19 Mai 2023).
- [35] PANDA, <https://pandas.pydata.org/docs/>, (accédé Le 19 Mai 2023).
- [36] NUMPY, <https://numpy.org/doc/stable/>, (accédé Le 19 Mai 2023).
- [37] MATPLOTLIB, <https://matplotlib.org/stable/index.html>, (accédé Le 19 Mai 2023).
- [38] OPENCV, https://docs.opencv.org/3.4/d0/de3/tutorial_py_intro.html, (accédé Le 19 Mai 2023).
- [39] TENSORFLOW, <https://www.tensorflow.org/?hl=fr>, (accédé Le 20 Mai 2023).
- [40] KERAS, <https://keras.io/about/>, (accédé Le 20 Mai 2023).
- [41] <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>, (accédé Le 20 Mai 2023)
- [42] Gridi Adel, Un Outil de Deep Learning pour les données textuelles, thèse MASTER ACADEMIQUE, Domaine : Mathématique et Informatique, Filière : Informatique,

Spécialité : Vision Artificielle, Faculté sciences exactes et sciences de la nature et de la vie, Département de mathématiques et d'informatique, Université LARBI ben M'HIDI O.E.B, 2019/2020.

- [43] [https //juripredis.com/blog/demystifier-le-machine-learning-partie-2-lesreseaux-de-neurones-artificiels/](https://juripredis.com/blog/demystifier-le-machine-learning-partie-2-lesreseaux-de-neurones-artificiels/) (accédé Le 20 Mai 2023).
- [44] <http://vision.gel.ulaval.ca/jflalonde/cours/4105/h16/tps/results/projet/BOFOC/index.html>, (accédé Le 20 Mai 2023).
- [45] Y. Zheng et E. Essock, A local-coloring method for night-vision colorization utilizing image analysis and fusion, *Information Fusion*, 2008.
- [46] A. Ami et al, Medical image colorization using optimization technique, *International Journal of Scientific and Research Publications*, 2013.
- [47] M. Martinez-Escobar, Colorization of CT images to improve tissue contrast for tumor segmentation, *International Journal of Scientific and Research Publications*, 2012.
- [48] Yu-Wing Tai et al, Local color transfer via probabilistic segmentation by expectation-maximization, In *Proc. CVPR*,(2005)
- [49] Jing Liao et al. Visual AttributeTransfer through Deep Image Analogy. *arXiv preprint arXiv*, (2017).
- [50] Alex Yong-Sang Chia et al, Semantic colorization with internet images, (*Proc. of SIGGRAPH Asia*) (2011).
- [51] Xiaopei Liu et al, Intrinsic colorization, (*Proc. of SIGGRAPH Asia*) (2008).
- [52] Aditya Deshpande et al, Learning large-scale automatic image colorization, . In *Proc. ICCV*, (2015).
- [53] Zezhou Cheng et al, Deep colorization. In *Proc. ICCV*, (2015).
- [54] Yi-Chin Huang et al, An adaptive edge detection based colorization algorithm and its applications, In *Proc. of the 13th annual ACM international conference on Multimedia*. ACM, (2005)
- [55] Yingge Qu et al, Manga colorization. *ACM Trans. Graph.* (*Proc. of SIGGRAPH Asia*), (2006)

- [56] J. D. Tascón et al, Deep Colorization of Medical Images Using Generative Adversarial Networks, 2018.
- [57] Yi-Chin Huang et al, An adaptive edge detection based colorization algorithm and its applications, international conference on Multimedia (2005)
- [58] Liron Yatziv and Guillermo Sapiro, Fast image and video colorization using chrominance blending. IEEE Trans. on Image Processing,(2006)
- [59] MARKLE, W., AND HUNT, B, Coloring a black and white signal using motion detection, 1988.
- [60] Phillip Isola et al, Image-to-image translation with conditional adversarial networks. In Proc. CVPR. 2017.
- [61] Guillaume Charpiat et al, Automatic image colorization via multimodal predictions, European conference (2008).